

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування та спеціалізованих

комп’ютерних систем

**Лабораторна робота №1**

**з дисципліни «Бази даних і засоби управління»**

Тема: **«***Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL***»**

Виконав:

студент групи КВ-03

Новіцький Віктор Миколайович

Перевірив:

Київ – 2023

Завдання роботи полягає у наступному:

1. Розробити модель «сутність-зв’язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».

2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.

3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).

4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**Приклади предметних галузей для проектування бази даних - «Школа»**

Обрана предметна галузь передбачає отримання і обробку даних про учнів та їхні телефони, вчителів та предмети школи. Згідно цієї області для побудови бази даних було виділено наступні сутності:

1. Учень (Student), з атрибутами: ім’я (First Name), прізвище (Last Name), номер телефону (Phone Number), ідентифікатор (student\_id). Призначена для того, щоб фіксувати учнів, які навчаються в школі.

2. Вчитель (Teachers), з атрибутами: ім’я (First Name), прізвище (Last Name), ідентифікатор (teacher\_id). Призначена для того, щоб фіксувати вчителів, які викладають у школі.

3. Предмет (Subject), з атрибутом назва (Name), ідентифікатор (subject\_id). Призначена для фіксування дисциплін, які викладаються у школі.

4. Розклад (Schedule), з атрибутами: день тижня (Day), час (Time), ідентифікатор (schedule\_id). Призначена для зв’язку між вчителями, предметами, які вони викладають, та учнями, що навчаються у даних вчителів на даному предметі. Також визначає день і час, коли проводиться урок.

5. Телефон (Phone), з атрибутами: ідентифікатор (phone\_id), номер телефону (phonenumber). Призначена для того, щоб визначати, який телефон має кожний учень.

**Модель «сутність-зв’язок» галузі «школа»**

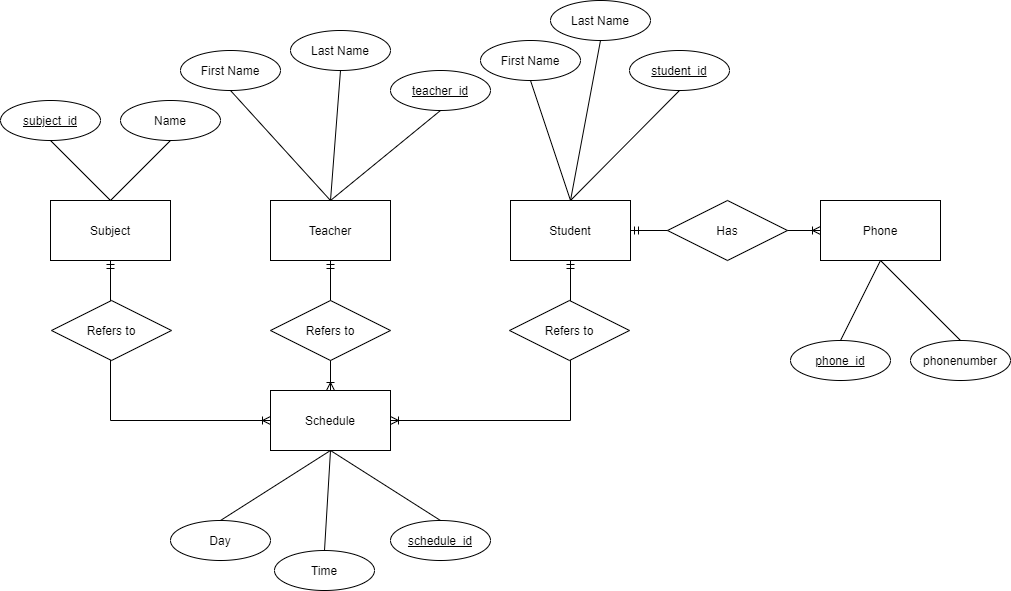


Рисунок 1. ER-діаграма, побудована за нотацією Чена

Модель «сутність-зв’язок» нарисована за допомогою інструменту <https://www.draw.io/>

**Опис зв’язків**

При проектуванні бази даних «Школа» можна виділити такі сутності. Учень (Student), Телефон (Phone), Вчитель (Teacher), предмет (Subject), розклад (Schedule).

Для реалізації зв’язку багато до багатьох (N:M) зумовлена поява таблиці Розклад (Schedule), так як ми зможем дізнатись інформацію, які дисципліни вивчаються кожним учнем (Student) і хто їх викладає (Teacher).

Кожний учень (Student) має декілька вчителів (Teacher) та предметів (Subject), тому щоб пов’язати ці сутності і визначити день та час, коли учень має певний урок, був встановлений зв’язок 1:N з таблицею розклад (Schedule).

Кожному учню (Student) належить один або декілька телефонів (Phone), тому щоб пов’язати ці сутності і визначити, кому який телефон належить, був встановлений зв’язок 1:N між сутностями Учень (Student) і Телефон (Phone).

Кожний вчитель (Teacher) може викладати декілька предметів (Subject) і навчати декількох учнів (Student), тому щоб пов’язати ці сутності і визначити день та час, коли викладач проводить певний урок, був встановлений зв’язок 1:N з таблицею Розклад (Schedule).

На кожному предметі (Subject) навчається декілька учнів (Student) і його може проводити декілька викладачів (Teacher), тому щоб пов’язати ці сутності і визначити день та час, коли відбувається урок, був встановлений зв’язок 1:N з таблицею Розклад (Schedule).

**Перетворення моделі у схему бази даних**

Сутність Учень (Student) була перетворена у таблицю Student з ідентифікатором student\_id і атрибутами firstname, lastname, phonenumber.

Сутність Викладач (Teacher) була перетворена у таблицю Teacher з ідентифікатором teacher\_id і атрибутами firstname, lastname.

Сутність Предмет (Subject) була перетворена у таблицю Subject з ідентифікатором subject\_id і атрибутом name.

Сутність Телефон (Phone) була перетворена у таблицю Phone з ідентифікатором phone\_id і атрибутом phonenumber.

Щоб реалізувати зв’язок багато до багатьох між таблицями Учень (Student), Викладач (Teacher), Предмет (Subject), була створена таблиця Розклад (Schedule), яка містить три зовнішніх ключа: subject\_fk для зв’язку з таблицею Предмет (Subject); student\_fk для зв’язку з таблицею Учень (Student); teacher\_fk для зв’язку з таблицею Викладач (Teacher); також ця таблиця містить атрибути Date та Time.

У розкладі кожний викладач (Teacher) має декілька учнів і предметів, тому щоб ідентифікувати, який викладач читає певний предмет певним учням, у таблиці Розклад (Schedule) маємо зовнішній ключ teacher\_fk.

У розкладі кожний предмет (Subject) має декілька учнів і викладачів, що його читають, тому щоб ідентифікувати, який предмет має певного викладача і учнів, у таблиці Розклад (Schedule) маємо зовнішній ключ subject\_fk.

У розкладі на кожному предметі (Subject) навчається декілька учнів, тому щоб ідентифікувати, який учень навчається на певному предметі і в певного вчителя, у таблиці Розклад (Schedule) маємо зовнішній ключ student\_fk.

Кожен учень (Student) може мати один або декілька телефонів, тому щоб ідентифікувати, якому учню належить певний телефон з певним номером, у таблиці Телефон Phone маємо зовнішній ключ studentphone\_fk.

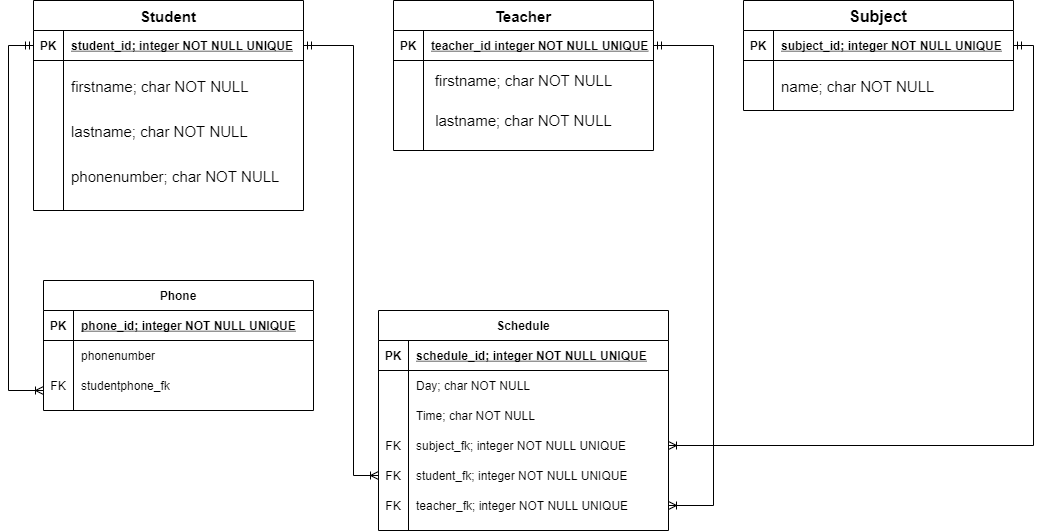


Рисунок 2 - ER-діаграма, переведена у таблиці БД

ER-діаграма нарисована за допомогою інструменту <https://www.draw.io/>

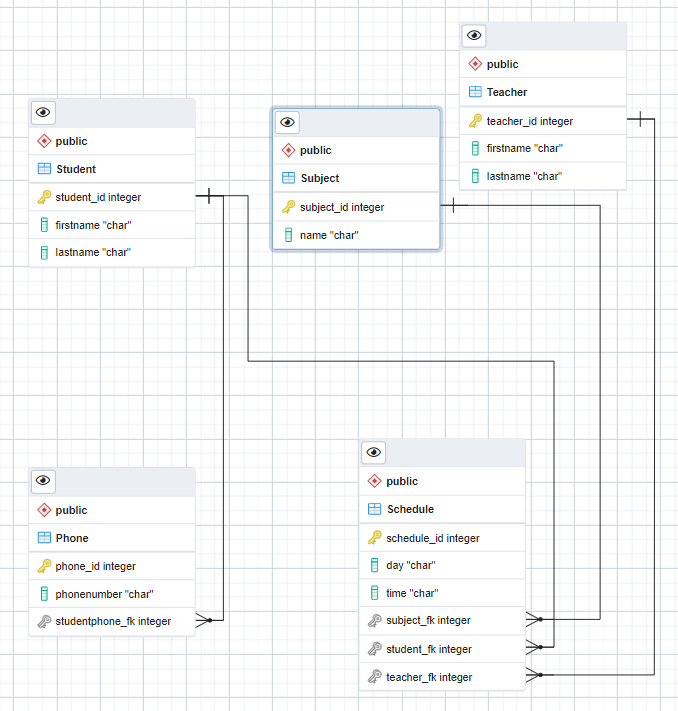


Рисунок 3. Схема бази даних у pgAdmin 4

Таблиця 1. Опис структури БД

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **Student**- *містить дані про учнів школи* | **student\_id** *- унікальний ідентифікатор учня*  **firstname** *- ім’я*  **lastname** *- прізвище учня* | **integer** *(числовий)*  **char** *(рядок)*  **char** *(рядок)* |
| **Teacher -** *містить дані про вчителів школи* | **teacher\_id -** *унікальний ідентифікатор вчителя*  **firstname -** *ім’я вчителя*  **lastname -** *прізвище вчителя* | **integer** *(числовий)*  **char** *(рядок)*  **char** *(рядок)* |
| **Subject** - *містить дані про шкільні предмети* | **subject\_id** - *унікальний ідентифікатор предмету*  **name** - *назва предмету* | **integer** *(числовий)*  **char** *(рядок)* |
| **Schedule –** *пов’язує предмети, учнів, що їх вивчають, і викладачів, що їх викладають.* | **schedule\_id –** *унікальний ідентифікатор розкладу*  **Day –** *день тижня, коли проводиться урок*  **Time –** *час, коли проводиться урок*  **subject\_fk –** *зовнішній ключ для зв’язку з предметом*  **student\_fk –** *зовнішній ключ для зв’язку з учнем*  **teacher\_fk –** *зовнішній ключ для зв’язку з вчителем* | **integer** *(числовий)*  **char** *(рядок)*  **char** *(рядок)*  **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |
| **Phone –** *містить дані про номери телефонів, що належать учням.* | **phone\_id** – *унікальний ідентифікатор телефону*  **phonenumber –** *номер телефону учня*  **studentphone\_fk –** *зовншній ключ для зв’язку з учнем* | **integer** *(числовий)*  **char** *(рядок)*  **integer** *(числовий)* |

**Функціональні залежності для кожної таблиці**

Student (*student\_id, firstname, lastname, phonenumber*):

student\_id → firstname

student\_id → lastname

student\_id → phonenumber

Teacher (*teacher\_id, firstname, lastname*):

teachers\_id → firstname

teachers\_id → lastname

teachers\_id → firstname, lastname

Subject (*subject\_id, name*):

subject\_id → name

Schedule (*schedule\_id, Day, Time, teacher\_fk, student\_fk, subject\_fk)*:

schedule\_id → Day

schedule\_id → Time

schedule\_id → teacher\_fk

schedule\_id → student\_fk

schedule\_id → subject\_fk

Phone (*phone\_id, phonenumber, studentphone\_fk)*:

phone\_id → phonenumber

phone\_id → studentphone\_fk

**Відповідність схеми бази даних до третьої нормальної форми**

Схема відповідає 1НФ, тому що в таблиці немає дубльованих рядків; у кожній комірці зберігається атомарне (скалярне) значення; у кожному стовпчику зберігаються дані одного типу.

Схема відповідає 2НФ, тому що вона відповідає 1НФ і має первинний ключ, а всі не ключові стовпчики таблиці залежать від первинного ключа.

Схема відповідає 3НФ, тому що вона відповідає 2НФ і всі неключові атрибути таблиці залежать винятково від усього первинного ключа, а не від його частини (тобто у схемі кожний неключовий атрибут нетранзитивно залежить від первинного ключа).

**Таблиці бази даних у pgAdmin4**

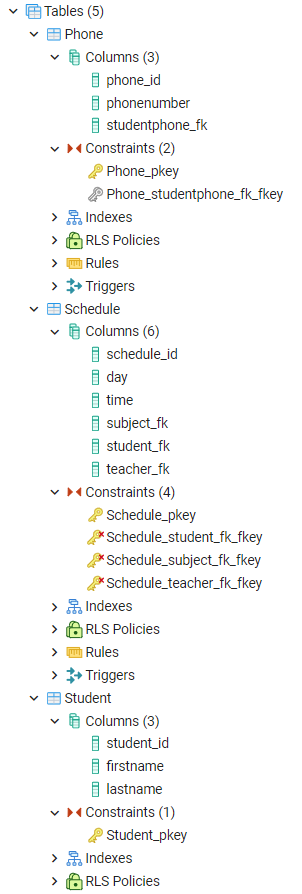


Рисунок 4.1. Всі таблиці в PgAdmin4

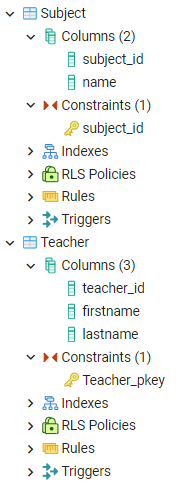


Рисунок 4.2. Всі таблиці в PgAdmin4

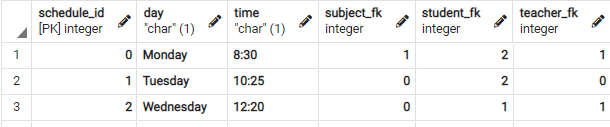


Рисунок 5. Дані таблиці Schedule



Рисунок 6. Дані таблиці Student

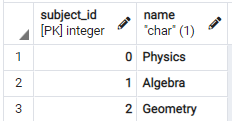


Рисунок 7. Дані таблиці Subject



Рисунок 8. Дані таблиці Teacher

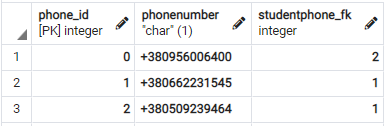


Рисунок 9. Дані таблиці Phone

**SQL-текст опису БД**

-- This script was generated by a beta version of the ERD tool in pgAdmin 4.

-- Please log an issue at https://redmine.postgresql.org/projects/pgadmin4/issues/new if you find any bugs, including reproduction steps.

BEGIN;

CREATE TABLE IF NOT EXISTS public."Phone"

(

phone\_id integer NOT NULL,

phonenumber "char" NOT NULL,

studentphone\_fk integer NOT NULL,

PRIMARY KEY (phone\_id)

);

CREATE TABLE IF NOT EXISTS public."Schedule"

(

schedule\_id integer NOT NULL,

day "char" NOT NULL,

"time" "char" NOT NULL,

subject\_fk integer NOT NULL,

student\_fk integer NOT NULL,

teacher\_fk integer NOT NULL,

PRIMARY KEY (schedule\_id)

);

CREATE TABLE IF NOT EXISTS public."Student"

(

student\_id integer NOT NULL,

firstname "char" NOT NULL,

lastname "char" NOT NULL,

PRIMARY KEY (student\_id)

);

CREATE TABLE IF NOT EXISTS public."Subject"

(

subject\_id integer NOT NULL,

name "char" NOT NULL,

PRIMARY KEY (subject\_id)

);

CREATE TABLE IF NOT EXISTS public."Teacher"

(

teacher\_id integer NOT NULL,

firstname "char" NOT NULL,

lastname "char" NOT NULL,

PRIMARY KEY (teacher\_id)

);

ALTER TABLE public."Phone"

ADD FOREIGN KEY (studentphone\_fk)

REFERENCES public."Student" (student\_id)

NOT VALID;

ALTER TABLE public."Schedule"

ADD FOREIGN KEY (student\_fk)

REFERENCES public."Student" (student\_id)

NOT VALID;

ALTER TABLE public."Schedule"

ADD FOREIGN KEY (subject\_fk)

REFERENCES public."Subject" (subject\_id)

NOT VALID;

ALTER TABLE public."Schedule"

ADD FOREIGN KEY (teacher\_fk)

REFERENCES public."Teacher" (teacher\_id)

NOT VALID;

END;